Energy & Buildings 216 (2020) 109952

Contents lists available at ScienceDirect

Energy & Buildings

journal homepage: www.elsevier.com/locate/enbuild

A long short-term memory artificial neural network to predict daily HVAC consumption in buildings

R. Sendra-Arranz*, A. Gutiérrez

E.T.S. Ingenieros de Telecomunicación, Universidad Politécnica de Madrid, Av. Complutense 30, Madrid 28040, Spain

ARTICLE INFO

Article history:
Received 26 April 2019
Revised 27 January 2020
Accepted 12 March 2020
Available online 19 March 2020

Keywords:
Energy consumption prediction
HVAC Systems
Short term forecast
Recurrent neural networks
LSTM layers

ABSTRACT

In this paper, the design and implementation process of an artificial neural network based predictor to forecast a day ahead of the power consumption of a building HVAC system is presented. The featured HVAC system is situated at MagicBox, a real self-sufficient solar house with a monitoring system. Day ahead prediction of HVAC power consumption will remarkably enhance the Demand Side Management techniques based on appliance scheduling to reach defined goals. Several multi step prediction models, based on LSTM neural networks, are proposed. In addition, suitable data preprocessing and arrangement techniques are set to adapt the raw dataset. Considering the targeted prediction horizon, the models provide outstanding results in terms of test errors (NRMSE of 0.13) and correlation, between the temporal behavior of the predictions and test time series to be forecasted, of 0.797. Moreover, these results are compared to the simplified one hour ahead prediction that reaches nearly optimal test NRMSE of 0.052 and Pearson correlation coefficient of 0.972. These results provide an encouraging perspective for real-time energy consumption prediction in buildings.

1. Introduction

Nowadays, there is an increasing concern in the society for the correct consumption and use of energy. This phenomenon has motivated the development of new technologies turning into real scientific challenges. One of the main problems in this context is to find a suitable equilibrium between the energy consumption and its generation. Unlike the ideal scenario of constant daily demand, the existence of load peaks of aggregated consumption in the activity hours and valleys with scarce demand (such as the night) produce undesired demand curves that fits badly with the energy generation. In short, valleys produce energy leaks (although it can be reduced with the use of storage strategies) because of the low demand. On the other hand, load peaks provoke unstable scenarios where demand surpasses instantaneous generation capabilities.

To solve these problems, Demand Side Management (DSM) strategies [1,2] have been studied and applied. Furthermore, in the recent years, Smart Grids [3,4] have emerged to aid the DSM purposes among other utilities. While the electrical grid was only meant to transport energy from producers to consumers, the Smart Grid focuses on analyzing users behavior and allows DSM techniques in conventional and distributed electric networks. Diverse DSM algorithms that harness the smart grid capabilities have been proposed. Most of them act as appliance controllers by means of operation scheduling with the aim to optimize one or several objectives, such as Peak-to-Average Ratio reduction, cost minimization or peak shaving. In [5], the authors propose a Generic DSM model, based on genetic algorithms, for residential users to reduce the Peak-to-Average Ratio, the total energy cost, and the waiting time of appliances. In addition, several residential load controlling techniques are described in [6]. It is based on the scheduling and time shifting of the operation of the loads in order to smooth the energy demand curve. It was shown how those methods reduce the energy consumption cost and the Peak-to-Average Ratio. Furthermore, in [7], the authors present simulated and real experiments integrating battery energy storage system (BESS) and Photovoltaic (PV) generation along with Active Demand-Side Management (ADSM) in a grid connected self sufficient house to maximize the PV energy self consumption. Finally, in [8], DSM techniques implemented by means of swarm intelligence are proposed.

However, the problem described above is too wide to be treated globally in detail. This paper focuses on the consumption of heating, ventilation and air-conditioning (HVAC) systems. In the USA, these systems represent more than 50% of the energy consumption in residential buildings and in China, a sample of 30 buildings exposed a 68% of residential consumption in average [9,10]. Moreover, knowing the future local consumption of the HVAC could allow demand response actions in grid connected systems or self-

* Corresponding author.
E-mail addresses: r.sendra@alumnos.upm.es (R. Sendra-Arranz), a.gutierrez@upm.es (A. Gutiérrez).

https://doi.org/10.1016/j.enbuild.2020.109952
0378-7788/

consumption actions if local generation is available. Nonetheless, the HVAC behaviour can be treated as a time series with certain periodicity. Therefore, to approach the time series prediction, different algorithms have been applied. For instance, linear regression models [11], autoregressive, moving average and autoregressive integrated moving average models [12], support vector machines [13,14] and artificial neural networks [15–18], among others.

In [11], a predictor of the power consumption of an HVAC system was implemented to aid the operation of two control strategies. These strategies aim to increase the PV self-consumption and grid-peak shaving respectively and were developed and assessed in the frame of a self-sufficient solar house, MagicBox, with integrated BESS, PV generation and monitoring systems. The HVAC predictor was based on a linear regression model and, as stated by the authors, the design of more accurate forecasting techniques was left as future research. The current paper can be seen as a continuation in that direction. Indeed, in this research, a more complex prediction system of the HVAC power consumption is designed, implemented and assessed under the same self sufficient house.

In order to design and implement a time series prediction model, two main factors should be taken into consideration: the algorithm that suitably fits the precise time series forecasting problem and the horizon of the predictions. Traditionally, the former task was mainly faced by means of linear regression [11] and time series analysis techniques. Regarding the latter approach, in [12], the authors compared autoregressive, moving average and autoregressive integrated moving average for short term load forecasting. Alternatively, machine learning algorithms have recently gained popularity due to the increase of the fitness of the results and their generalization capabilities (among many other advantages). Support vector machines are an example of a machine learning technique harnessed to address the time series prediction problem [13,14,19]. Additionally, artificial neural networks (ANN) have been used in [15,16]. Moreover, the use of recurrent neural networks (RNN) is strongly recommended because they are able to retain and consider the temporal variations of the time series throughout their feedback connections. More precisely, HVAC power consumption time series denote high periodicity, mainly because of the daily and seasonal periodic nature of ambient variables and human habits. This fact justifies the use of RNN as the prediction model in the context of HVAC power consumption forecasting. In [17], the authors take advantage of RNNs to develop a model to forecast hourly energy consumption. Finally, in [20], a wavelet packet decomposition is applied to a wind speed time series. Afterwards, a one dimension convolutional neural network (non recurrent) is used to predict the higher frequency components and a convolutional neural network placed before a Long-Short Term Memory (LSTM) [18] neural network forecasts the lower frequency variations. The predictions for one step to three steps ahead are compared with other predictors. Additionally, in [21], the authors use an LSTM neural network for petroleum production forecasting. In the research presented in this manuscript, LSTM neural networks are used.

Subsequently, the horizon of the predictions has to be chosen according to the problem requirements. Short and long term predictions are the most known time horizons. As aforementioned, the predictor designed here is thought to aid the DSM techniques such as scheduling loads. Therefore short term forecasting is the desirable time horizon for this application. In [22,23], the authors compare different prediction models to forecast short term load consumption. In addition, the authors of [24] addressed the HVAC load short-term prediction problem by means of support vector regression and ANN. The model was preceded by a preprocessing stage based on correlation analysis, principal component analysis and wavelet decomposition of the data. Moreover, in [25] a support vector regression trained by means of a genetic algorithm (GA-SVR) with wavelet decomposition is established to address short term and ultra short term in a similar HVAC load prediction context. However, as the prediction models in [24,25] are not dynamic, they do not consider the periodicity of the forecasted time series. The usage of systems with memory states could enhance the accuracy of the predictions. In contrast, when the application requires larger prediction horizons, long term forecasts should be considered. An example of long term forecasting is studied in [26] with the use of an encoder–decoder based LTSM ANN.

The LSTM neural networks proposed in this paper have been trained using the backpropagation through time algorithm [27]. Moreover, the Adam optimizer [28] is proposed as the learning wrapper that remarkably enhances the training process. Adam optimizer has been widely used in the literature when dealing with deep learning architectures [20,26] due to its fast convergence, little hyper-parameter tuning and adaptive learning rates for each parameter using the first and second moment estimates of the gradient.

The main contribution of this paper is the development and verification of several models to perform short term forecasts of the power consumption of HVAC systems in buildings. The predictor model consists of a stacked LSTM ANN trained by several data obtained from the aforementioned self-sufficient solar house MagicBox. The models will output the next day consumption prediction based on the previous day behaviour, being able to retain its temporal dependencies by means of the dynamic nature of the LSTM ANN. This leads to an enhancement in the precision of the predictions. More precisely, the three presented models show strong performance results highlighting test Pearson correlation coefficients around 0.797 and normalized root mean square error (NRMSE) of 0.13 for the most accurate model. Additionally, one hour ahead predictions are separately performed and compared to those mentioned before. The prediction of the next hour of power consumption leads to outstanding Pearson correlation coefficient of 0.972 and NRMSE of 0.052 at the cost of reducing the horizon of the forecasts. These results provide an encouraging perspective for real-time energy consumption prediction in buildings.

The remainder of this paper is as follows: Section 2 exposes the theoretical preliminaries on recurrent neural networks that will be used throughout this paper. Section 3 is devoted to the design process. The nature of the utilized dataset is commented, and the aforementioned self-sufficient house, where the data was measured, is introduced. Afterwards, the preprocessing and arrangement techniques for the data set are described. Finally, the main architecture of the RNN models and the three multi step prediction ahead architectures are presented and analyzed. Section 4 deals with the implementation environment and the predictor hyperparameter optimization. In Section 5 the assessment of the designed models is performed. Finally, Section 6 concludes the paper.

2. Theoretical preliminaries

The main tools utilized in the paper approach are artificial neural networks (see [29–31]). Indeed, as it is a time series prediction problem, recurrent neural networks (RNN) [32] are selected; note that this sort of networks excels when the treated problem involves input and output data with sequential nature.

Within this scenario, let $\mathbf{x}_k$ and $\mathbf{h}_k^{(m)}$ be the input and output vectors at time step k, and let $\mathbf{h}_{k-1}^{(m)}$ be the output vector at the previous instant. In addition, let $\mathbf{\Phi}^{(m)}$ be the activation function of the mth layer and let $\mathbf{W}$ and $\mathbf{U}$ be the weight matrices that apply a linear transformation to $\mathbf{x}_k$ and $\mathbf{h}_{k-1}^{(m)}$, respectively. Finally, let $\mathbf{b}$ be a bias vector, that will be used to apply an affine transformation to the linear transformation $\mathbf{W}$ and $\mathbf{U}$. Then, Eq. (1) describes the

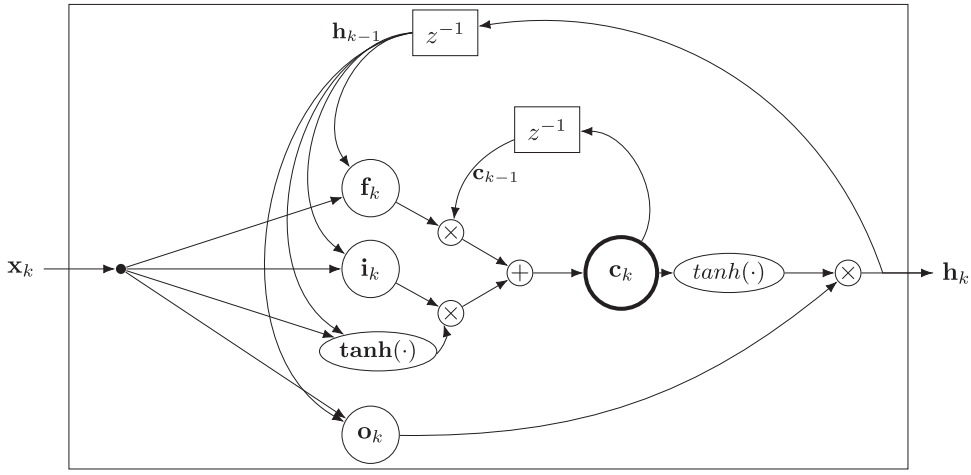


Fig. 1. Computational graph of an LSTM layer. **tanh(·)** denotes the application of *tanh*(·) to the entries of $\mathbf{W}_c\mathbf{x}_k + \mathbf{U}_c\mathbf{h}_k + \mathbf{b}_c$ and *tanh*(·) applies an hyperbolic tangent to the entries of the cell state.

behaviour of a multilayer recurrent neural network,

$$\begin{cases} \mathbf{h}_k^{(m)} &= \mathbf{\Phi}^{(m)}\left(\mathbf{W}^{(m)}\mathbf{h}_k^{(m-1)} + \mathbf{U}^{(m)}\mathbf{h}_{k-1}^{(m)} + \mathbf{b}^{(m)}\right) \\ \mathbf{h}_k^{(1)} &= \mathbf{\Phi}^{(1)}\left(\mathbf{W}^{(1)}\mathbf{x}_k + \mathbf{U}^{(1)}\mathbf{h}_{k-1}^{(1)} + \mathbf{b}^{(1)}\right) \end{cases} \tag{1}$$

where $m \in \{2, \ldots, M\}$ and M is the number of layers.

The models presented in this paper have been trained by means of the backpropagation algorithm [33], and more precisely the backpropagation through time (BPTT) variant [27]. BPTT is a learning algorithm that extends the backpropagation method to train RNNs, and is based on the unfolding of the RNN to convert the recurrent architecture into a multilayer perceptron. Thereafter, the regular backpropagation method can be applied to the unfolded RNN. For further details on backpropagation and BPTT algorithms, refer to [29–31]. Furthermore, in order to improve the results of these algorithms, the Adam optimizer is used (see [28]).

A typical disadvantage of the BPTT is the vanishing gradient phenomenon that usually happens when the number τ, of time steps used to unfold the RNN, is too large. In order to overcome this difficulty we use a particular type of RNN, namely Long short-term memory (LSTM) neural networks [18]. LSTM neural networks are able to maintain both short and long time dependencies through its states. It is composed by gates that control the information that is stored in the states by filtering the input and output flows of data. These gates are called forget, input and output gates. An LSTM unit contains two different states, the hidden state ($\mathbf{h}_k$) and the cell state ($\mathbf{c}_k$). The former is analogous to the state of a regular RNN described in Eq. (1). This state represents the output value of the LSTM layer set by the output gate. In contrast, the cell state ($\mathbf{c}_k$) is the actual memory that is controlled by the forget and input gates.

Eq. (2) states the set of computations performed by an LSTM layer with multiple units. In addition, the computational graph of the formulas is depicted in Fig. 1.

$$\left.\begin{aligned} \mathbf{f}_k &= \sigma\left(\mathbf{W}_f\mathbf{x}_k + \mathbf{U}_f\mathbf{h}_{k-1} + \mathbf{b}_f\right) \\ \mathbf{i}_k &= \sigma\left(\mathbf{W}_i\mathbf{x}_k + \mathbf{U}_i\mathbf{h}_{k-1} + \mathbf{b}_i\right) \\ \mathbf{o}_k &= \sigma\left(\mathbf{W}_o\mathbf{x}_k + \mathbf{U}_o\mathbf{h}_{k-1} + \mathbf{b}_o\right) \\ \mathbf{c}_{i_k} &= \mathbf{i}_k \circ \tanh\left(\mathbf{W}_c\mathbf{x}_k + \mathbf{U}_c\mathbf{h}_{k-1} + \mathbf{b}_c\right) \\ \mathbf{c}_k &= \mathbf{f}_k \circ \mathbf{c}_{k-1} + \mathbf{c}_{i_k} \\ \mathbf{h}_k &= \mathbf{o}_k \circ \tanh(\mathbf{c}_k) \end{aligned}\right\} \tag{2}$$

In both, Eq. (2) and Fig. 1 $\mathbf{f}_k$, $\mathbf{i}_k$ and $\mathbf{o}_k$ are the forget, input and output gates respectively. σ and *tanh* represent the sigmoid and the hyperbolic tangent activation functions and $\circ$ denotes the Hadamard product.

Fig. 2. Frontal view of the MagicBox. The MagicBox is located at the *ETSI* de Telecomunicación of the *Universidad Politécnica de Madrid*.

3. Design process

3.1. Data acquisition system

In order to develop the training process of the ANN through the backpropagation algorithm, a dataset, containing the power consumption records, and several inputs with a remarkable correlation with the consumption, is required. To assess this task, a dataset was extracted from a real solar house called http://www.magicbox.etsit.upm.es/ MagicBox.

The MagicBox is a self-sufficient solar house (see Fig. 2) that integrates sustainable elements based on renewable energies, self-sufficiency energetic methods, bioclimatic architecture and recycled construction materials [34]. In addition, it includes Information and Communication Technologies to monitor and control the house power flow. It includes PV generation, electricity storage through batteries, a set of automated appliances and a connection to the grid [35]. The PV installation consists of single-crystalline PV generators that are distributed in four south-oriented surfaces with different inclinations. The energy is collected in six arrays with a total nominal power of $7.2kW_p$. In addition, the electrical system embodies a battery energy storage system of $36kWh$. In a grid-connected installation, batteries are used to improve the electrical behavior by controlling the maximum consumed and generated

power at different hours or ensuring the electrical supply when a grid breakdown occurs.

MagicBox includes typical electrical appliances of a highly electrified home: washing machine, dryer, dishwasher, refrigerator, cooking appliances, lighting, computers and entertainment appliances. The appliances are integrated in a home automation system, which allows them to be monitored and controlled by a remote system [36]. Some appliances involve an instantaneous use because of the user demand (e.g. lights, TVs, computers) while others can be time-shifted.

MagicBox was originally designed to participate in the Solar Decathlon 2005 contest (see [37,38]). Since then, multiple studies have been developed on it. In [39] an heterogeneous collaborative sensor network designed to manage the energy performance of the MagicBox was described. In addition, [36] presented the operation of a semi-distributed electrical demand-side management system with the PV generation in order to improve the self-consumption. Moreover, the optimization of the self-consumption in a system with the PV generation coupled to a battery energy storage system and connected to the grid was studied and tested in [11]. Finally, in [8] a swarm intelligence approach has been implemented for Demand-Side Management. In this paper, only the HVAC system and the sensors that measure the different environmental variables were utilized.

Due to the nature of the problem, the input dataset is strongly related to the weather. On the one hand, outdoor measurement variables such as the outdoor temperature [$^\circ$ C], the relative humidity [%] and the irradiance [$\frac{W}{m^2}$] measurements are considered. These input variables will provide the machine learning system with an insight of the physical behaviour of the weather. On the other hand, three indoor variables are included: the indoor CO_2 level, the indoor temperature of the house and the reference temperature set by the user to fit the desired indoor comfort temperature. Ideally, the indoor temperature and the reference temperature should match. However, the dynamics of the MagicBox and the outdoor climate conditions will prevent the indoor temperature to exactly follow the reference value. It should be pointed out that there were not occupants in the MagicBox during the monitored period. In conclusion, the RNN will receive the 6 aforementioned input variables and will output the forecasted power consumption. This output variable will be provided to the RNN in the format of target examples of the desired output values during the learning process. The distributed nodes related to this paper are located in the electrical box and are directly connected to the temperature, humidity, CO_2, power meters and the meteorological station. Temperature, humidity and CO_2 sensors are based on the SCR110-H from Schneider Electric with a real resolution of 0.1° C for temperature, 0.1% for humidity and 1ppm for CO_2. Electric meters are based on the iEM3155 from Schneider Electric. Additionally, the HVAC equipment (see [11]) is a two air-to-air electrical reversible heat pumps from Daikin, corresponding to model FTX25KV1B. Its main advantages, that have produced an important market penetration, are the high efficiency of the heating and cooling processes and the fact that both functionalities are joined in the same installation. The switch between cooling and heating can be readily performed by inverting the operating cycle with a reversing valve.

In addition to the sensors, a real-time distributed monitoring system is installed in the MagicBox for its continuous supervision. The monitoring system is based on an ad-hoc designed embedded system connected through an RS-485 serial bus to several distributed nodes. The nodes are running a real-time operating system which takes care of acquiring all measurements at specific time frames. Every 10 ms, each node obtains information from all the sensors to which it is connected. Every second, the monitoring node performs the mean of the last 100 measurements and stores it internally to be acquired by the main controller through

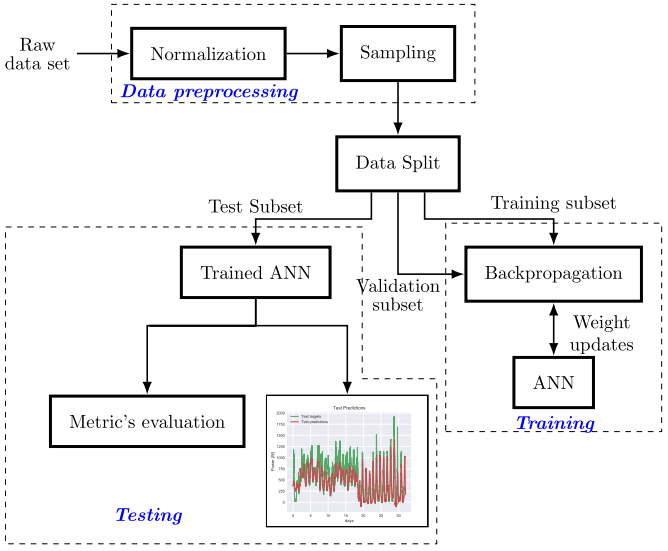


Fig. 3. Diagram of the design process.

an RS-485 RTU-Modbus protocol. The main controller is based on an embedded Linux operating system. It is made of a microcontroller with a UPS system and an electronic carrier to connect to the nodes. Every minute, the main controller acquires all measurements stored on the nodes and accumulates them on an internal SD card. This information is stored together with the date and time. Moreover, every five minutes the main controller sends the data stored to the monitoring server. The monitoring server is in charge of synchronizing all data, performing all conversions from electrical to physical measurements and running periodic scripts which process the data and upload them to the monitoring database. A monitoring website has been created to allow researchers and the general public observe the behavior of the MagicBox in real time. All the information presented in this manuscript is extracted from that website and is accessible to the general public. The data obtained to the analysis performed is gathered from July 2016 until September 2019, although monitoring is still ongoing. The time series dataset is currently composed by more than 1.5 million of samples with 1 min sampling period.

3.2. Data preprocessing

The data, as mentioned in Section 3.1, was obtained from several sensors by an automated data acquisition system. This measured data is in a raw format. Thus, in order to train the RNN, an appropriate set of preprocessing techniques should be applied to the original data. Fig. 3 describes the steps and data flow followed during the training and test processes. More precisely, the normalization, sampling and data split blocks are the data preprocessing stages described in this section.

The RNN receives, as input, r time series ($r = 6$) of data, and generates, as output, s time series ($s = 1$); each time series is a vector of length ℓ (representing the number of samples). It should be pointed out that the RNN also receives the power consumption in the previous day as input to forecast the next day. Therefore, in practice, r is actually 7. For superindex $i \in \{1, \ldots, r+s\}$,

$$\mathbf{x}_k^{(i)} = \begin{bmatrix} x_1^{(i)} \\ \vdots \\ x_\ell^{(i)} \end{bmatrix} \tag{3}$$

represents the corresponding time series, and subindex k denotes the discrete time variable, with a sampling period of 1 min. This

notation will be used in subsequent sections as well. Fig. 3 shows a description of the data preprocessing process.

The first preprocessing technique considered is data normalization. The dataset normalization selected consists of transforming the original range of the time series into a range between 0 and 1. More precisely, the preprocessing works as follows: let $\mathbf{x}_i^{\min}$ denote the minimal value that $\mathbf{x}_i(k)$ takes when the time k moves from 1 to ℓ. That is,

$$\mathbf{x}_{\min}^{(i)} = \min\{x_k^{(i)} \mid k \in \{1, \dots, \ell\}\}.$$

Similarly,

$$\mathbf{x}_{\max}^{(i)} = \max\{x_k^{(i)} \mid k \in \{1, \dots, \ell\}\}.$$

In this situation, the time series $\mathbf{x}_k^{(i)}$ is normalized as

$$\widetilde{\mathbf{x}}_k^{(i)} = \frac{\mathbf{x}_k^{(i)} - \mathbf{x}_{\min}^{(i)}}{\mathbf{x}_{\max}^{(i)} - \mathbf{x}_{\min}^{(i)}} \tag{4}$$

In addition to the normalization technique, a data sampling process is proposed. The main motivation is that the original one minute sampling period of the data becomes unnecessary for the problem solving. Therefore, in this paper, an increase of 15 min is stated. The corresponding value of the new time series will be the mean of the values of $\mathbf{x}^{(i)}$ in the enlargement period, so that the information within the 15 min is not lost. More precisely, the following variables are considered: the original discrete time variable k (see Eq. (3)), which takes values from 1 to ℓ minutes, and a new time variable, named k^*, which takes values from T to nT minutes, where nT is the greater multiple of T being smaller than ℓ; in our case $T = 15$ min. So, $k = Tk^*$. Then, the new time series is defined as

$$\mathbf{z}_{k^*}^{(i)} = \frac{1}{T} \sum_{k=(k^*-1)T}^{k^*T} \mathbf{x}_k^{(i)}, \quad \mathbf{z}_1^{(i)} = \mathbf{x}_1^{(i)} \tag{5}$$

The selected sampling period T in the multi step architectures was set to 15 min. This resolution allows to reduce the number of samples in the prediction of a day, from 1440 to 96. In order to simplify notations, in the subsequent text, we denote $\mathbf{x}_k$ as the downsampled and normalized time series vector.

Finally, the original dataset $\mathcal{A}_{\text{data}}$ is divided into three different subsets. $\mathcal{A}_{\text{data}}$ is

$$\mathcal{A}_{\text{data}} = (\mathcal{A}_{\text{train}} \cup \mathcal{A}_{\text{val}}) \cup \mathcal{A}_{\text{test}},$$

where the first two subsets are devoted to the training and dynamical validation of the training respectively, while the third subset is used to test the final result of the training process. In addition, due to the sequential nature of our problem, it is crucial that the mentioned subsets contain ordered data to preserve the temporal behaviour of the time series. In order to train and test the system, the size of the described subsets were fixed to 80% for $\mathcal{A}_{\text{train}}$, 5% for $\mathcal{A}_{\text{val}}$, and 15% for $\mathcal{A}_{\text{test}}$.

3.3. Neural architecture

This subsection is devoted to introduce the recurrent neural network used to forecast the consumption of the HVAC system. The generic model will be adapted and optimized along the remaining sections, and all the subsequent models described throughout this paper are extensions, or modifications, of this architecture.

Let H be the prediction horizon ($H = 96$), that corresponds to one day prediction. The main structure of the proposed model (see Fig. 4) is formed by two LSTM layers (L_1 and L_2), with N_1 LSTM units in L_1 and N_2 in L_2, followed by a perceptron layer to output the corresponding prediction (the selected values for N_1 and N_2 are shown in Section 4). L_1 and L_2 consist of the LSTM computational graph described in Section 2 (see Fig. 1).

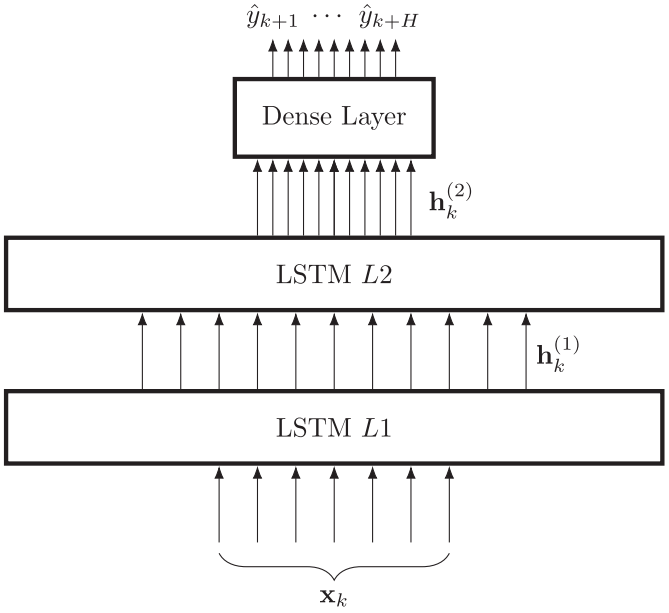


Fig. 4. Main structure of the proposed model. LSTM layers are building blocks containing LSTM neural networks and dense layer represents a single layer perceptron.

To be more precise, let $\mathbf{X}$ denote the matrix

$$\mathbf{X} = (\mathbf{x}_k^{(1)}, \dots, \mathbf{x}_k^{(7)}) = \begin{bmatrix} x_1^{(1)} & \cdots & x_1^{(7)} \\ \vdots & \ddots & \vdots \\ x_\ell^{(1)} & \cdots & x_\ell^{(7)} \end{bmatrix} \tag{6}$$

where $\mathbf{x}_k^{(i)}$ is as in Eq. (3). Furthermore, let $\mathbf{x}_k$ denote the kth row of $\mathbf{X}$, that is

$$\mathbf{x}_k = \left[x_k^{(1)}, \cdots, x_k^{(7)}\right]^T \tag{7}$$

L_1 receives $\mathbf{x}_k$ as input to generate $\mathbf{h}_k^{(1)}$ as output (see Eq. (2) and Fig. 1). $\mathbf{h}_k^{(1)}$ is a vector whose length equals the number of units in the layer (N_1). This new vector $\mathbf{h}_k^{(1)}$ is then taken as input of L_2. Afterwards, L_2 generates a new vector, $\mathbf{h}_k^{(2)}$, with length is N_2. Finally, the non-recurrent fully-connected layer gets $\mathbf{h}_k^{(2)}$ to derive the power consumption predictions $\hat{y}_{k+1}, \dots, \hat{y}_{k+H}$.

3.4. Prediction techniques

The main objective of the RNN is to make short term multi step predictions of the consumption. More precisely, the model will take previous data in order to forecast the next day. The following formula describes the main behaviour of multi step prediction:

$$[\hat{y}_{k+1}, \hat{y}_{k+2}, \dots, \hat{y}_{k+H}] = \hat{f}(\mathbf{x}_k, \mathbf{h}_{k-1}^{(1)}, \mathbf{c}_{k-1}^{(1)}, \mathbf{h}_{k-1}^{(2)}, \mathbf{c}_{k-1}^{(2)}) \tag{8}$$

In this subsection, we consider different modifications, of the main RNN described above, to obtain different multi step predictors. These models are designed using unfolding techniques of the RNN so that a deeper understanding of the temporal behavior is achieved.

First multi step prediction approach: MSPM-1

The first approach, referred to as Multi Step Prediction Model 1 (MSPM-1), is to forecast all the values between $k+1$ and $k+H$ at once. Therefore, at each time step, the RNN will receive the observed input data at that precise time instant and will output the estimated evolution of the time series for the next 24 h. This model will require H neurons in the output layer,where H is the

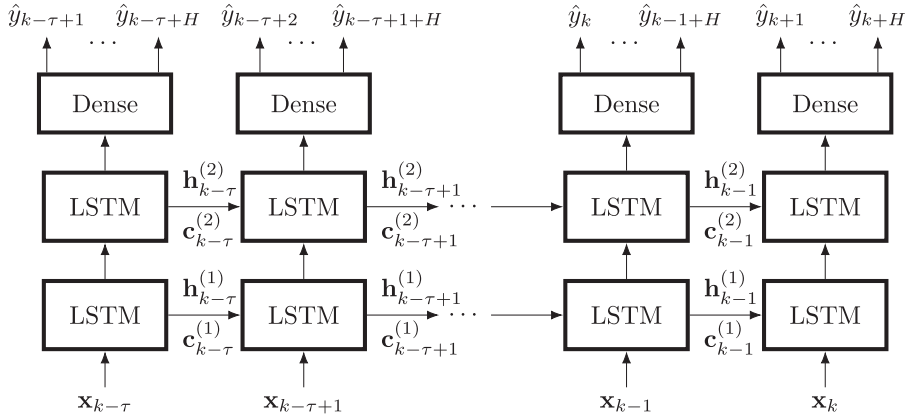


Fig. 5. Unrolling of the MSPM-1.

number of future predictions to be forecasted. Thus, each neuron in this non-recurrent layer is responsible of the estimation of a single time step of the power consumption. The day ahead prediction is formed by the union of all the outputs of the layer neurons.

Fig. 5 shows the unrolling of MSPM-1. The variable τ represents the number of past time steps to be considered during the training stage to perform backpropagation through time algorithm. Notice that large values of τ would produce stronger vanishing of the gradient. Alternatively, in the evaluation and deployment stages, the predictions are performed in an online way. Thus, in these cases, τ time steps should be elapsed in order to make the first forecast. This unavoidable issue is caused by the fact that the predictor requires at least a sequence of τ samples in order start performing the estimations. In addition, it should be mentioned that the model provides not only the value at $k+H$ time step but also predicts the evolution of the time series between k and $k+H$. These values may be useful depending on the precise DSM strategy to be used afterwards. Therefore, the subsequent prediction model, proposes some modifications to cover those cases that only require the point estimation $\hat{y}_{k+H}$.

Second multi step prediction approach: MSPM-2

The second proposed model complements the previous one so that, at each time step k it only provides the prediction of y_{k+H}. In fact, it is direct to realize that the predictor MSPM-1 is a generalization of MSPM-2. Therefore, it is supported on a simpler structure that can be used in order to accomplish the short term prediction when just the estimate $\hat{y}_{k+H}$ is needed. In some cases, the time series evolution between k and $k+H$ discrete instants can be approximated by an interpolation using these extreme values.

Fig. 6 illustrates MSPM-2. In this architecture the output perceptron layer consists of a single neuron. Unlike in MSPM-1, this predictor uses a multilayer perceptron with 2 layers. This is a hidden layer with 20 neurons and the output layer with a single neuron.

Third multi step prediction approach: MSPM-3

Finally, the MSPM-3 is presented. This approach is based on the encoder–decoder ANN proposed in [40], as well as on the sequence–to–sequence neural network described in [41]. It has been widely used in scenarios that require the transformation between a sequence in one domain to a sequence in another domain. Some examples are the field of machine translation, such as in [42], or multi-step time series prediction problems [26] as the one treated in this work. The architecture is composed by two basic recurrent layers: the encoder or reader and the decoder or writer. The encoder is responsible of transforming an input sequence of data into a fixed-length vector representing the whole sequence, namely, the context ($\mathbf{z}_k$). In contrast, the decoder receives the context produced by the encoder and generates the decoded sequence, which is indeed the output sequence. More precisely, the decoder uses the latent representation $\mathbf{z}_k$ as the initial state for its dynamics.

Fig. 7 represents the architecture of the unfolded encoder–decoder neural network. It can be noticed that in MSPM-3, both encoder and decoder are LSTM layers. The feedforward dense layer adapts the decoded sequence to generate the actual prediction. Moreover, the prediction is used as the input in the next time step of the decoding process. The input of the decoder at the first decoding instant is the true current value of the power consumption y_k It should be noticed that even though Fig. 7 only depicts the decoding of the context at discrete instant k, this decoding procedure is performed every time step in order to obtain all the predictions between $k+1$ and $k+H$.

4. Implementation environment and predictor hyper-parameter optimization

In order to implement the designed models, a programming environment and a hyper-parameter selection process has been fixed. With respect to the programming environment, all the models were implemented, trained and tested using the Python's library Pytorch [43]. The code related to the paper was uploaded to a Github repository[1].

The chosen hyper-parameters to be studied have been: the number of units and the learning rate α. These are some of the most relevant features of the RNN structures and the learning algorithm. On the one hand, the number of units should be large enough to provide the RNN with an adequate computational power. In addition, an excess of parameters could produce overfitting with large training times. On the other hand, the learning rate is responsible of enabling a proper convergence in the backpropagation algorithm as fast as possible. Another important decision to be taken involves the learning opimizer. In all the models, Adam [28] optimizer was fixed. Some alternatives to Adam optimizer are stochastic gradient descent and RMSprop [44]. However, due to its adaptive learning rate, fast convergence and little hyper-parameter tuning, Adam is the most used optimizer for deep learning architectures in the literature [20,26].

Boxplot representation were used to depict the empirical process. The x axis of these plots represents the studied hyper-parameter, and the y axis shows the distribution of the absolute error of the predictions defined as $\left|y_{k+H} - \hat{y}_{k+H}\right|$ with $k \in \{\tau, \dots \ell -$

[1] https://github.com/Robolabo/LSTM-HVAC

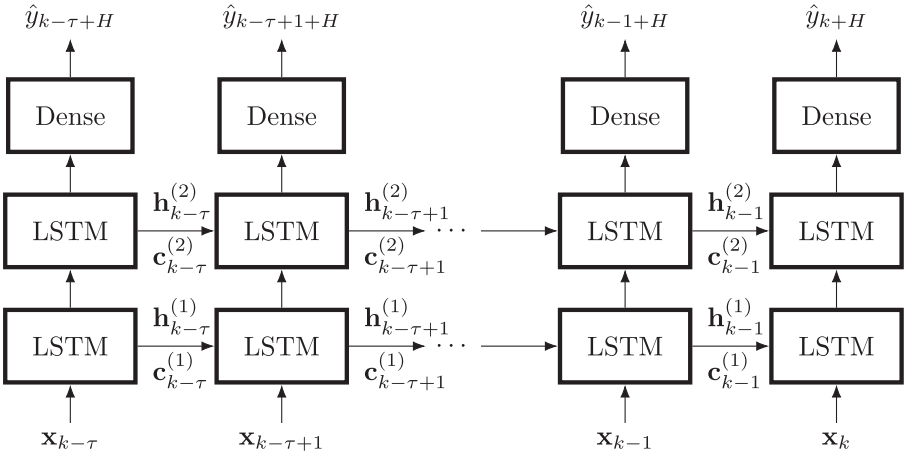


Fig. 6. Unrolling of the MSPM-2.

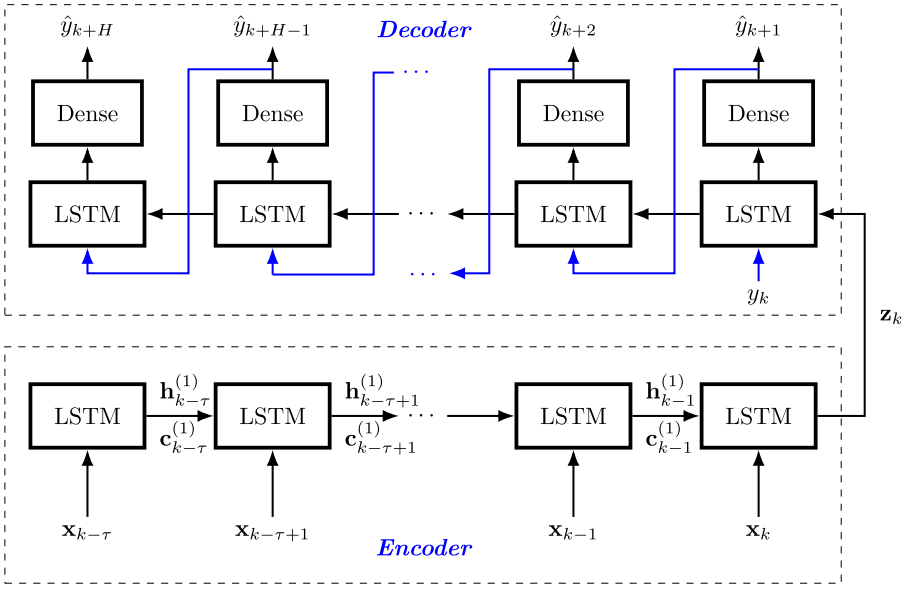


Fig. 7. Unrolling of the MSPM-3.

$H\}$, where ℓ is the sample size of the test data subset. Thus each box shows the distribution of the absolute error of the H step ahead predictions for the hyper-parameter to be assessed.

$$\mathbf{e} = \left(\left|y_{\tau+H} - \hat{y}_{\tau+H}\right|, \ldots, \left|y_{i+H} - \hat{y}_{i+H}\right|, \ldots, \left|y_{\ell} - \hat{y}_{\ell}\right|\right)$$

for the corresponding model.

Fig. 8 a, compares the impact that the number of units produces when using the different MSPMs. For simplicity, only the models with $N_1 = N_2$ are depicted. In addition, Fig. 8 b, shows the impact of the learning rate (α) within the designed models. A learning rate of $\alpha = 0.001$, and *Adam* optimizer were used when assessing the number of units. Additionally, 30 units in both layers and Adam optimizer were fixed in order to evaluate the impact of α on the prediction. In both scenarios, a τ of 96 past time steps, corresponding to an entire day, was applied.

Fig. 8 a shows that at least 20 units are required to provide the RNN with enough computational power. On the contrary, the predictors with more than 20 units per layer produce a highly similar absolute error distribution. Therefore, for simplicity, the number of units per LSTM layer was fixed to 20. Indeed, a large number of units would imply longer training times and could produce overfitting.

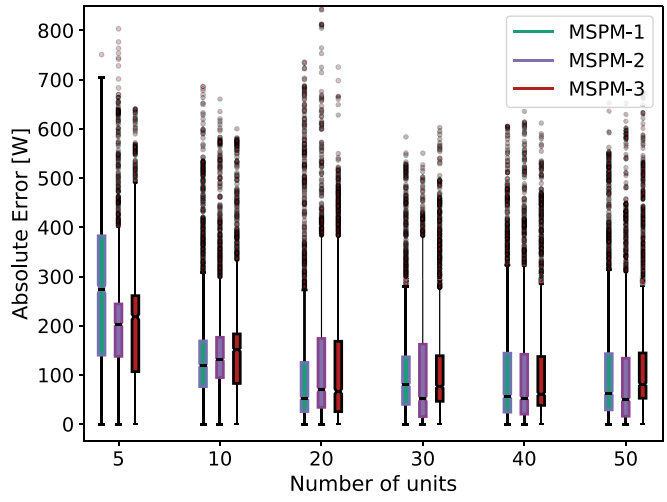
Table 1
Summary of the selected hyper-parameters for the MSMPs.

τ	Layers	Units	Optimizer	α
96	2	32	Adam	0.001

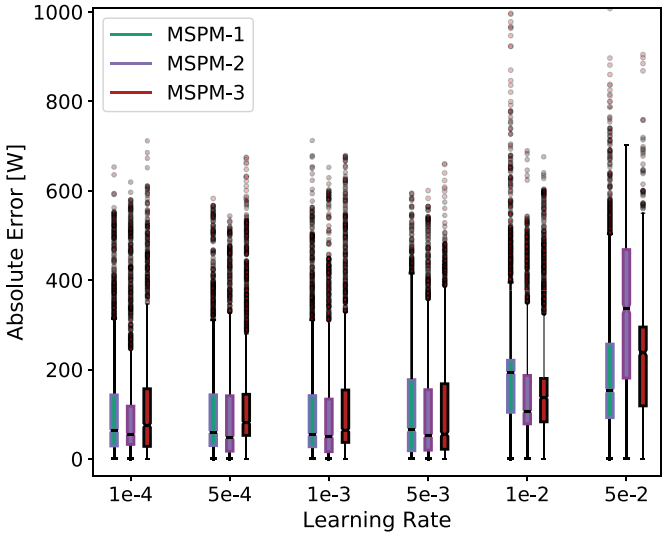
In Fig. 8 b, it can be appreciated how, for large values, the predictions lead to very high errors. For values lower than 0.005, the absolute errors considerably decrease and distributions tend to be remarkably similar. A value of $\alpha = 0.001$ was selected for all the models in order to obtain a tradeoff between training speed and stability. Notice that other values of α could have been a feasible choice as well. Table 1 collects the selected hyper-parameters for all the prediction models.

5. Results

This section is devoted to the performance evaluation analysis of the proposed models. For this purpose, several metric comparisons will be studied. Moreover, different figures will be displayed to illustrate the accuracy of the estimations. The training and testing process of the models use the hyper-parameters highlighted in



(a)



(b)

Fig. 8. Comparison of models: number of units in (a) and the learning rate (α) in (b). Each boxplot comprises observations ranging from the first to the third quartile. The median is indicated by a horizontal bar, dividing the box into the upper and lower parts. The whiskers extend to the farthest data points that are within 1.5 times the interquartile range. Outliers are shown with a red points. A value of $\alpha = 1 \cdot 10^{-3}$ was used to train the models in (a) and a 30 units in both layers were used in the case of (b). All the assessed predictors used $\tau = 96$ corresponding to the previous day and were trained with Adam optimizer. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 1. Moreover, a downsampling of 15 min is carried out by following the guidelines described in Section 3. Firstly, the LSTM architectures were trained in order to address the 24 h ahead forecast (see Section 3 for architectural details). Finally, the treated 24 h ahead prediction is compared to the 1 h ahead estimations.

In the evaluation stage, the trained predictors are sequentially fed with the $l - H$ testing input samples. Thereafter, each predictor produces a set of estimations $\hat{\mathbf{y}} = \left[\hat{\mathbf{y}}_{\tau+H}, \ldots \hat{\mathbf{y}}_l\right]^T$. It should be pointed out that the first $\tau + H$ predictions are not considered because the models require τ input samples to proceed and the forecast is H steps ahead. Secondly, although some predictors estimate the whole sequence between k and $k+H$, for a proper comparison of the models, the assessment was applied only to the estimations of $k+H$ at each discrete instant. This decision diverges from the training phase, in which the minimization of the cost function considers the sample mean of all the output predictions. With this information in mind, the following assessment metrics will be used:

- The mean squared error (MSE)

$$\mathrm{MSE}(\hat{\mathbf{y}}, \mathbf{y}) = \frac{1}{l - \tau - H} \sum_{k=\tau+H}^{l} (\hat{\mathbf{y}}_k - \mathbf{y}_k)^2 \tag{9}$$

- The root mean squared error (RMSE):

$$\mathrm{RMSE}(\hat{\mathbf{y}}, \mathbf{y}) = \sqrt{\frac{1}{l - \tau - H} \sum_{k=\tau+H}^{l} (\hat{\mathbf{y}}_k - \mathbf{y}_k)^2} \tag{10}$$

- The normalized root mean squared error (NRMSE):

$$\mathrm{NRMSE}(\hat{\mathbf{y}}, \mathbf{y}) = \sqrt{\frac{1}{l - \tau - H} \sum_{i=\tau+H}^{l} \left(\frac{\hat{\mathbf{y}}_k - \mathbf{y}_k}{\mathbf{y}_{\max}}\right)^2} \tag{11}$$

- The Pearson correlation coefficient ($\rho_{y,\hat{y}}$):

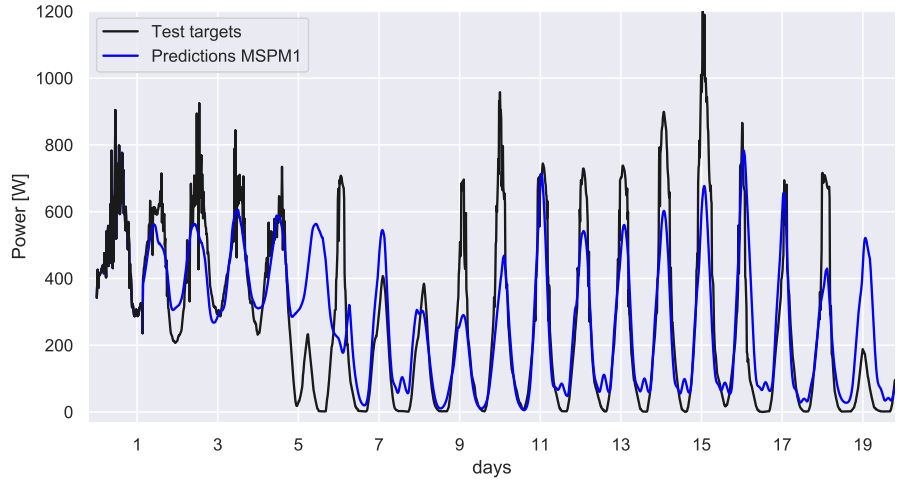
$$\rho_{\mathbf{y},\hat{\mathbf{y}}} = \frac{E[(\mathbf{y} - \mu_{\mathbf{y}})(\hat{\mathbf{y}} - \mu_{\hat{\mathbf{y}}})]}{\sigma_{\mathbf{y}} \sigma_{\hat{\mathbf{y}}}} \tag{12}$$

where the expected value was approximated by the sample mean of $l - \tau - H$ samples. The evaluation of the previously defined metrics applied to all the models is arranged in Table 2. Although it was briefly commented before, all the metrics and figures in this section denote the fitness of the H steps ahead predictions.

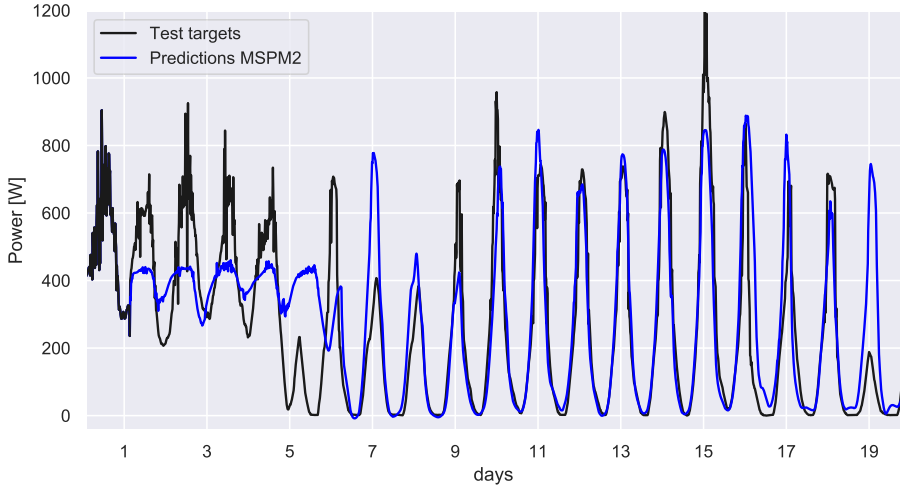
Fig. 9 shows the response of MSPM-1, MSPM-2 and MSPM-3 against the 24 h ahead prediction problem when the test data is provided. In both Fig. 9 and Table 2 it can be distinguished that the predictions of all the proposed models provide similar results in terms of errors, correlation and curve fitting. MSPM-2 slightly outstands the other models with a NRMSE of 0.130 and correlation coefficient of 0.797 at the cost of predicting just the H-step ahead value. On the contrary, even though MSPM-1 and MSPM-2 produce inferior estimations in terms of errors and correlation coefficient, they can be used in those DSM strategies that require the consumption during the entire posterior day.

In Fig. 9, it can be observed that the segment of time series to be forecasted follows two models with noticeably different means and variances. This issue imposes the modelling of a time series that behaves differently depending on the discrete time instant (the time series is highly non-stationary). In fact, MSPM-1 and MSPM-3 encounter some problems when estimating the first part of the testing sequence and provide proficient predictions when the second window of the time series is addressed. On the contrary, MSPM-2 is able to approximate more accurately the power consumption at the beginning of the evaluation but is not able to fully capture the amplitude of all the peaks in the second part of the sequence. In addition. it is utterly interesting to analyse the predictions when the time series behavior is switched (around day 5). In this case, it can be highlighted that there is a transient period of time that the LSTM models require in order to realize the behavioral modifications of the environment and adapt their forecasts with the aid of the LSTM inner state dynamics.

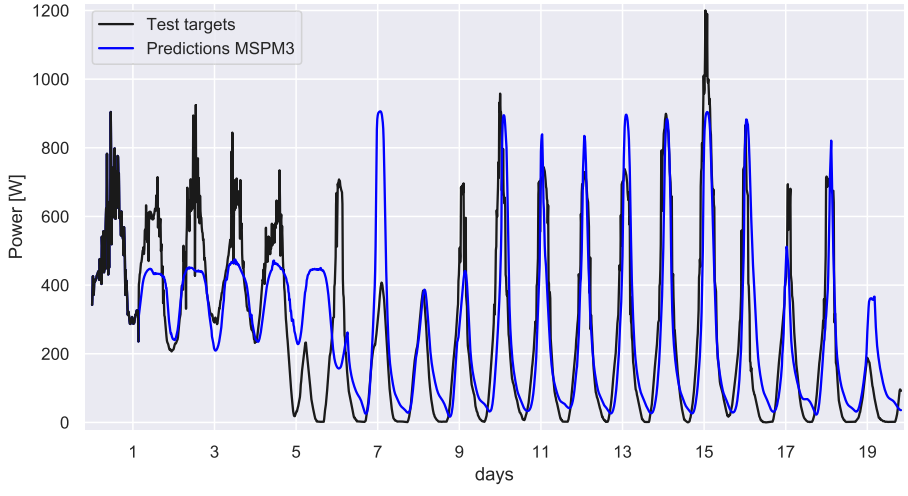
To conclude with the verification of the models, a comparison between the aforementioned output time series is described. Fig. 10 depicts the correlation between the predicted and the real time series for all the treated models. At the top of the figures, the power consumption distribution of the target time series can be observed. On the right side, the distribution of the predicted power consumption appears. The dark purple areas represent a high density of points and clearer purple areas denote a low density of samples. The figures support the Pearson correlation coefficient exposed in Table 2. Although there is margin for improvement, the graphs illustrate a remarkable linear relationship between ground



(a)



(b)



(c)

Fig. 9. Assessment of MSPM-1 (a), assessment of MSPM-2 (b) and assessment of MSPM-3 (c) when test dataset is provided. As the predictions are 24 h ahead with 15 min of time resolution, the valid predictions start to be generated at $k = \tau + 96$. Although MSPM-1 and MSPM-3 jointly estimate all the values between $k+1$ and $k+96$, only the estimates $\hat{y}_{k+96}$ are depicted.

Table 2
Collection of all the model's metrics for training and test data set. The shown MSE values correspond to the normalized targets and predictions time series.

	Training				Test			
Model	MSE	RMSE[W]	NRMSE	$\rho_{y,\hat{y}}$	MSE	RMSE[W]	NRMSE	$\rho_{y,\hat{y}}$
MSPM-1	$8.23 \cdot 10^{-3}$	113.39	0.094	0.795	0.0175	158.75	0.135	0.789
MSPM-2	$8.77 \cdot 10^{-3}$	110.24	0.091	0.821	0.0170	157.13	0.130	0.797
MSPM-3	$9.95 \cdot 10^{-3}$	117.55	0.113	0.788	0.0174	157.89	0.131	0.782

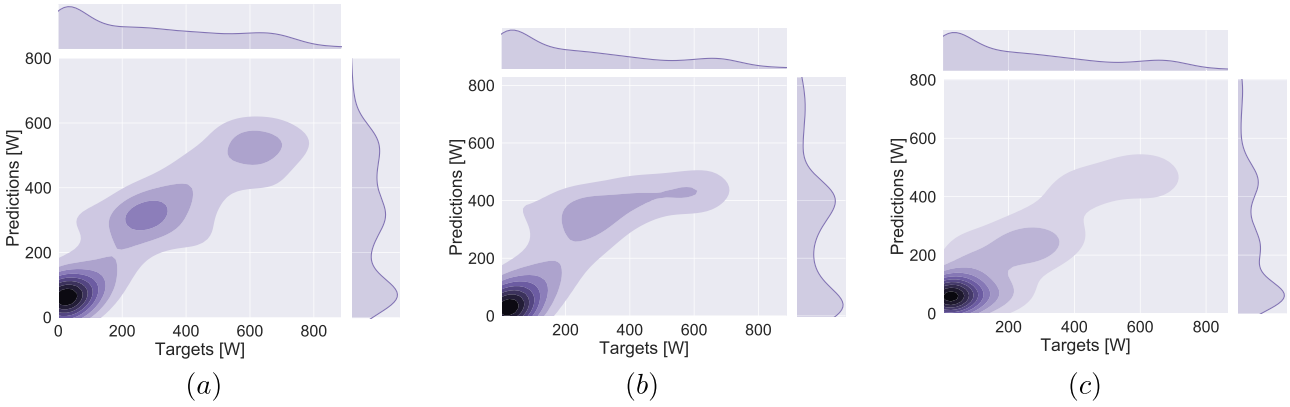


Fig. 10. Representation of the correlation between the model's predictions and the target time series. (a) MSPM-1's predictions. (b) MSPM-2's predictions. (c) MSPM-3's predictions. The darker means a higher density of samples in an area and clearer regions denote a low density of samples. Although MSPM-1 and MSPM-3 jointly estimate all the values between $k+1$ and $k+96$, only the estimates $\hat{y}_{k+96}$ are depicted.

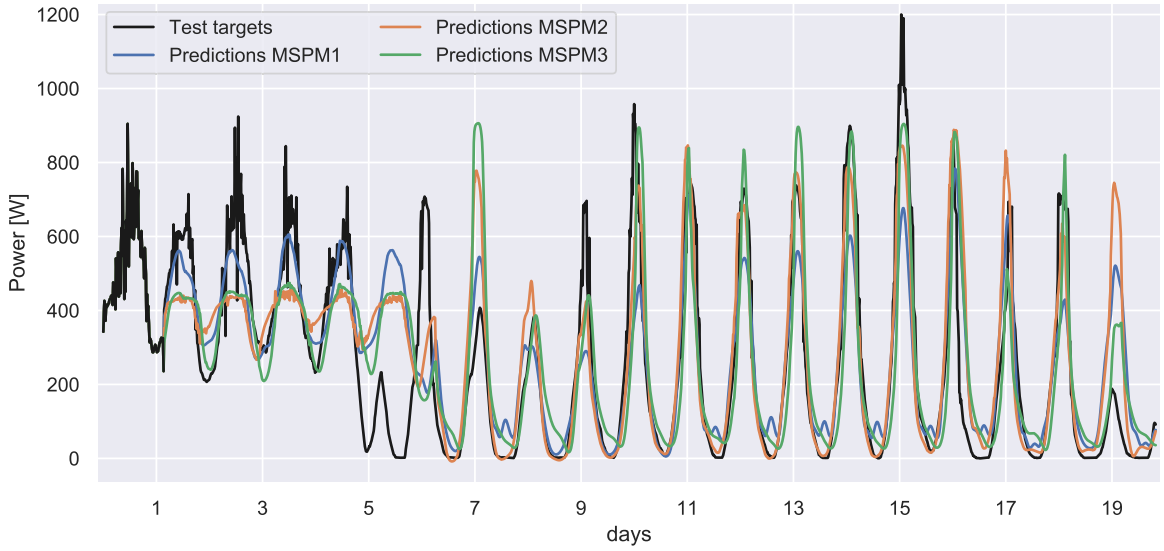


Fig. 11. Comparison of the predictions made by the models when the test data set is provided. As the predictions are 24 h ahead with 15 min of time resolution, the valid predictions start to be generated at $k = \tau + 96$. Although MSPM-1 and MSPM-3 jointly estimate all the values between $k+1$ and $k+96$, only the estimates $\hat{y}_{k+96}$ are depicted.

truth time series and estimated time series. Notice that a perfect linear relationship would imply that $|\rho_{\mathbf{y},\hat{\mathbf{y}}}| = 1$.

Fig. 11 jointly depicts the 24 h ahead predictions of all the models in a single graph for a clearer comparison of the forecasts.

Moreover, Fig. 12 shows the predictions of the models paying attention to a four day window to deeply appreciate the fitting of the time series. MSPM-2 generates outstanding predictions, being able to mainly capture the target time series with low delay. On the contrary, MSPM-3 produces delayed responses only in the rising intervals. However, the delay is still much lower than the prediction horizon. MSPM-1 exposes a balanced behavior, with predictions that are not highly delayed with respect to the targets but are unable to fully capture the peak consumption.

Finally, the predictors have been separately trained for the one step ahead forecasting problem. The aim is to compare the impact of the prediction horizon on the estimations. Fig. 13 shows the 1 h step ahead predictions of all the proposed models when test subset is provided. The fitness of the estimations in this case is nearly optimal. Moreover, in some intervals, the responses are denoised versions of the original time series to be forecasted.

Additionally, Table 3 gathers the analysed metrics applied to the test subset in the frame of the one hour ahead prediction problem. MSPM-2 is slightly superior than the other predictors, highlighting a Pearson correlation coefficient of 0.972 and NRMSE and RMSE of 0.052 and 62.47 respectively.

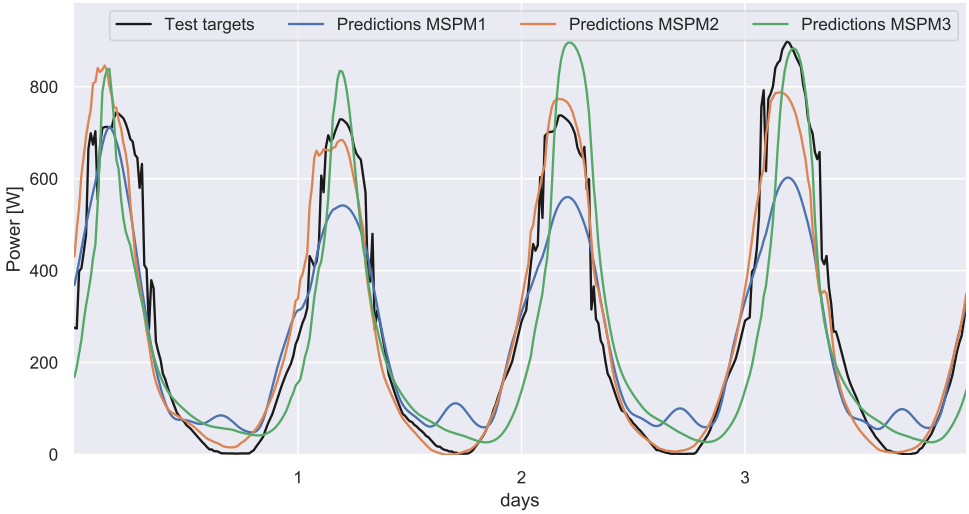


Fig. 12. Predictions of the models in a window of 4 days. Although MSPM-1 and MSPM-3 jointly estimate all the values between $k+1$ and $k+96$, only the estimates $\hat{y}_{k+96}$ are depicted.

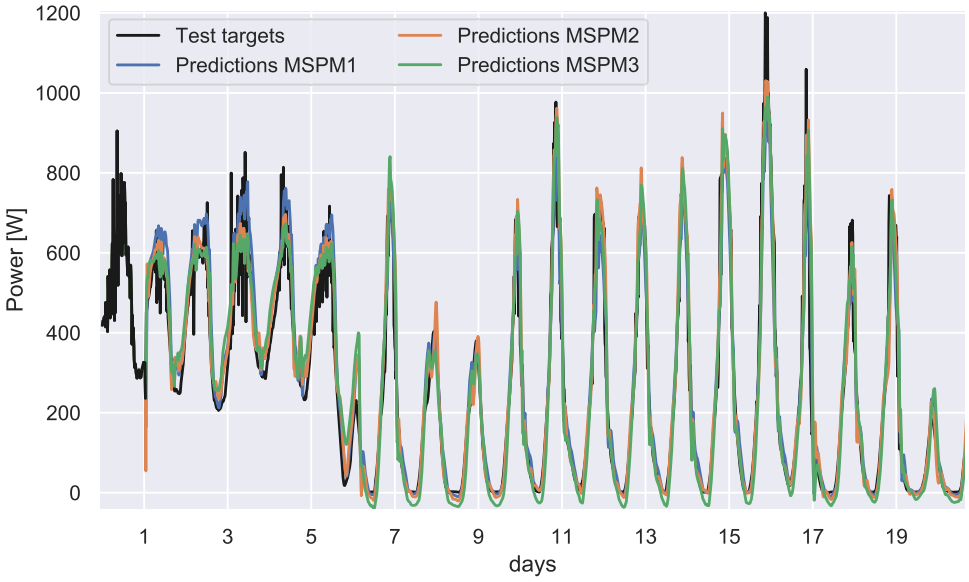


Fig. 13. One hour ahead predictions of the models when test subset is provided. As the predictions are 1 h ahead with 15 min of time resolution, the valid predictions start to be generated at $k = \tau + 4$. Although MSPM-1 and MSPM-3 jointly estimate all the values between $k+1$ and $k+4$, only the estimates $\hat{y}_{k+4}$ are depicted.

Table 3
Collection of all the model's metrics for test data when predicting one hour ahead. The shown MSE values correspond to the normalized targets and predictions time series.

Model	MSE	RMSE[W]	NRMSE	$\rho_{y,\hat{y}}$
MSPM-1	$3.17 \cdot 10^{-3}$	67.25	0.055	0.965
MSPM-2	$2.7 \cdot 10^{-3}$	62.47	0.052	0.972
MSPM-3	$3.16 \cdot 10^{-3}$	67.42	0.056	0.966

The results obtained in the 1 h ahead prediction are superior to those in the case of the 24 h forecast. This performance degradation is feasible due to the fact that the prediction horizon is much higher. On the one hand the use of the former forecasting scheme will allow the planning of remarkably accurate DSM strategies in very near future scenarios. On the other hand, the latter predictor allows the development of longer term DSM scheduling at the cost of having higher uncertainty due to the increased error in the predictions. Depending on the precise DSM application requirements, a tradeoff between the one hour ahead and one day ahead must be considered.

6. Conclusion and future lines

The forecast of the power consumed by an HVAC system located in a self sufficient solar house was the addressed problem. The house, called http://www.magicbox.etsit.upm.es/ MagicBox and located at the *Escuela Técnica Superior de Ingenieros de Telecomunicación* (ETSIT) of the *Universidad Politécnica de Madrid* (UPM), is equipped with a monitoring system to acquire the data. The main goal was to predict the next day of the power time series given the previous day to feed a future demand-side management system. To accomplish this task several prediction models were proposed. These models are RNNs based on LSTM layers to capture the sequential nature of the time series. The designed RNNs were implemented with the ANN Python's library Pytorch. The model with highest accuracy of the predictions reaches a test Pearson correlation coefficient of 0.797 and normalized root mean square

error (NRMSE) of 0.13. In addition to the 24 h ahead prediction, a simplified problem of hourly prediction is treated. With the reduction of the prediction horizon the models generate estimations that lead to outstanding results supported by a test Pearson correlation coefficient of 0.972 and a normalized root mean square error (NRMSE) of 0.052. A balance between error of the forecasts and prediction horizon must be fixed by the requirements of the DSM strategies that harness the proposed models. During the training and testing process, it was observed that the input variables CO_2 and relative humidity have little impact on the performance of the models. This issue was not deeply explored, and thus, a rigurous study of importance of each input variable towards predicting the power consumption of HVAC systems may enhance the accuracy of the estimations.

Other research lines can be proposed as future work. Firstly, the integration of the developed predictor into a distributed DSM system to be controlled as a deferable load is the main extension to this work to be considered in the future. Subsequently, the improvement of the predictors to gain accuracy in the 24 h ahead forecast or the increase of the prediction horizon to longer term scheduling can be studied. The development of a non-Markovian multi-step ahead predictor to properly model time series that do not satisfy the Markov property is an appealing topic. More precisely, it would receive not only the hidden and cell states at the current iteration but also the states at $k-2,\ldots,k-L$. Additionally, the mixture between convolutional neural networks and LSTM networks in a single architecture or the use of temporal convolutional networks can be addressed. Other non-architectural ideas to increase the performance of the models is to accomplish more efficient hyper-parameter tuning strategies such genetic algorithms or Bayesian optimization or, as already mentioned, to assess the importance of each variable in the process of predicting the power consumption. Furthermore, the exploration of time series data augmentation with generative adversarial networks is also an important future research line to considered. Finally, a better solution to deal with the behavioral changes in the time series exposed in Section 5 can be analyzed. Differentiation of the power consumption time series aiming to convert it into a stationary process can be assessed in order to solve the issue. Alternatively, the creation of a pool of prediction models and an algorithm that decides which predictor in that set should be used depending on the time interval should also be explored.

Declaration of Competing Interest

We wish to draw the attention of the Editor to the following facts which may be considered as potential conflicts of interest and to significant financial contributions to this work. [OR] We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

We confirm that the manuscript has been read and approved by all named authors and that there are no other persons who satisfied the criteria for authorship but are not listed. We further confirm that the order of authors listed in the manuscript has been approved by all of us.

We confirm that we have given due consideration to the protection of intellectual property associated with this work and that there are no impediments to publication, including the timing of publication, with respect to intellectual property. In so doing we confirm that we have followed the regulations of our institutions concerning intellectual property.

We understand that the Corresponding Author is the sole contact for the Editorial process (including Editorial Manager and direct communications with the office). He/she is responsible for communicating with the other authors about progress, submissions of revisions and final approval of proofs. We confirm that we have provided a current, correct email address which is accessible by the Corresponding Author and which has been configured to accept

Acknowledgments

This work was partially supported by the “DEMS: Sistema Distribuido de Gestión de Energía en Redes Eléctricas Inteligentes”, funded by the Programa Estatal de Investigación Desarrollo e Innovación orientada a los retos de la sociedad of the Spanish Ministerio de Economía y Competitividad (TEC2015-66126-R). Rafael Sendra acknowledges support from the Spanish Ministry of Education, culture and sports under Collaboration grant.

References

[1] P. Palensky, D. Dietrich, Demand side management: demand response, intelligent energy systems, and smart loads, IEEE Trans. Ind. Inf. 7 (3) (2011) 381–388.
[2] G. Strbac, Demand side management: benefits and challenges, Energy Policy 36 (12) (2008) 4419–4426.
[3] M.L. Tuballa, M.L. Abundo, A review of the development of smart grid technologies, Renew. Sustain. Energy Rev. 59 (2016) 710–725.
[4] P. Siano, Demand response and smart grids–a survey, Renew. Sustain. Energy Rev. 30 (2014) 461–478.
[5] K.M. Asghar, J. Nadeem, M. Anzar, K.Z. Ali, A. Nabil, A generic demand side management model for smart grid, Int. J. Energy Res. 39 (7) (2015) 954–964.
[6] M.N. Ullah, N. Javaid, I. Khan, A. Mahmood, M.U. Farooq, Residential energy consumption controlling techniques to enable autonomous demand side management in future smart grid communications, in: 2013 Eighth International Conference on Broadband and Wireless Computing, Communication and Applications, 2013, pp. 545–550.
[7] M. Castillo-Cagigal, E. Caama-Martín, E. Matallanas, D. Masa-Bote, A. Gutiérrez, F. Monasterio-Huelin, J. Jiménez-Leube, PV self-consumption optimization with storage and active DSM for the residential sector, Sol. Energy 85 (9) (2011) 2338–2348.
[8] M. Castillo-Cagigal, E. Matallanas, E. Caamaño-Martín, Álvaro Gutiérrez, Swarmgrid: demand-side management with distributed energy resources based on multifrequency agent coordination, Energies 11 (9) (2018) 1–16.
[9] L. Pérez-Lombard, J. Ortiz, C. Pout, A review on buildings energy consumption information, Energy Build. 40 (3) (2008) 394–398.
[10] R. Jing, M. Wang, R. Zhang, N. Li, Y. Zhao, A study on energy performance of 30 commercial office buildings in hong kong, Energy Build. 144 (2017) 117–128.
[11] J. Solano, L. Olivieri, E. Caama-Martín, Assessing the potential of PV hybrid systems to cover HVAC loads in a grid-connected residential building through intelligent control, Appl. Energy 206 (2017) 249–266.
[12] J. Deng, P. Jirutitijaroen, Short-term load forecasting using time series analysis: a case study for singapore, in: 2010 IEEE Conference on Cybernetics and Intelligent Systems, 2010, pp. 231–236.
[13] Z. Hou, Z. Lian, An application of support vector machines in cooling load prediction, in: 2009 International Workshop on Intelligent Systems and Applications, IEEE, Wuhan, China, 2009, pp. 1–4.
[14] E. Ceperic, V. Ceperic, A. Baric, A strategy for short-term load forecasting by support vector regression machines, IEEE Trans. Power Syst. 28 (4) (2013) 4356–4364.
[15] M. Beccali, M. Cellura, V.L. Brano, A. Marvuglia, Short-term prediction of household electricity consumption: assessing weather sensitivity in a mediterranean area, Renew. Sustain. Energy Rev. 12 (8) (2008) 2040–2065.
[16] D.C. Park, M.A. El-Sharkawi, R.J. Marks, L.E. Atlas, M.J. Damborg, Electric load forecasting using an artificial neural network, IEEE Trans. Power Syst. 6 (2) (1991) 442–449.
[17] P.A. Gonzlez, J.M. Zamarreo, Prediction of hourly energy consumption in buildings based on a feedback artificial neural network, Energy Build. 37 (6) (2005) 595–601.
[18] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[19] B. Dong, C. Cao, S.E. Lee, Applying support vector machines to predict building energy consumption in tropical region, Energy Build. 37 (5) (2005) 545–553.
[20] H. Liu, X. Mi, Y. Li, Smart deep learning based wind speed prediction model using wavelet packet decomposition, convolutional neural network and convolutional long short term memory network, Energy Convers. Manage. 166 (2018) 120–131.
[21] A. Sagheer, M. Kotb, Time series forecasting of petroleum production using deep LSTM recurrent networks, Neurocomputing 323 (2019) 203–213.
[22] P. Lusis, K.R. Khalilpour, L. Andrew, A. Liebman, Short-term residential load forecasting: impact of calendar effects and forecast granularity, Appl. Energy 205 (2017) 654–669.
[23] K. Gajowniczek, T. Zabkowski, Short term electricity forecasting using individual smart meter data, Procedia Comput. Sci. 35 (2014) 589–597.

[24] Y. Ding, Q. Zhang, T. Yuan, K. Yang, Model input selection for building heating load prediction: a case study for an office building in Tianjin, Energy Build. 159 (2018) 254–270.
[25] Y. Ding, Q. Zhang, T. Yuan, Research on short-term and ultra-short-term cooling load prediction models for office buildings, Energy Build. 154 (2017) 254–267.
[26] A. Rahman, V. Srikumar, A.D. Smith, Predicting electricity consumption for commercial and residential buildings using deep recurrent neural networks, Appl. Energy 212 (2018) 372–385.
[27] P.J. Werbos, Backpropagation through time: what it does and how to do it, Proc. IEEE 78 (10) (1990) 1550–1560.
[28] D. Kingma, J. Ba, Adam: a method for stochastic optimization., arXiv preprint arXiv:1412.6980(2014).
[29] S.S. Haykin, Neural Networks and Learning Machines, third ed., Pearson Education, Upper Saddle River, NJ, 2009.
[30] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016. http://www.deeplearningbook.org
[31] H.B. Demuth, M.H. Beale, O. De Jess, M.T. Hagan, Neural Network Design, second ed., Martin Hagan, USA, 2014.
[32] R.D. Beer, J.C. Gallagher, Evolving dynamical neural networks for adaptive behavior, Adapt. Behav. 1 (1) (1992) 91–122.
[33] D. E. Rumelhart, G. E. Hinton, R. J. Williams, Learning representations by back-propagating errors, Nature 323 (1986) 533–536.
[34] A. Gutiérrez, J. Jiménez-Leube, L. Magdalena, A distributed sensor network for the control of a bioclimatic house in Spain, Sensors 9 (10) (2009) 8197–8214.
[35] M. Castillo-Cagigal, E.C. no Martín, A. Gutiérrez, D. Masa-Bote, F. Monasterio, J. Porro, E. Matallanas, J.-J. Leube, Self-consumption of PV electricity with active demand side management: the GeDELOS-PV system, in: Proceedings of the 25th European Photovoltaic Solar Energy Conference, WIP-Renewable Energies, Munich, Germany, 2010, pp. 4866–4870.
[36] M. Castillo-Cagigal, A. Gutiérrez, F. Monasterio-Huelin, E. Caamaño-Martín, D. Masa-Bote, J. Jiménez-Leube, A semi-distributed electric demand-side management system with PV generation for self-consumption enhancement, Energy Convers. Manage. 52 (7) (2011) 2659–2666.
[37] E. Caamamartn, M.A. Egido, J. Neila, C. Bedoya, A.G. Santos, F.J. Jimnez, L. Magdalena, Spanish participation in the solar decathlon 2005 competition: new proposals for zero-energy houses, in: Proceedings of the 20th European Photovoltaic Solar Energy Conference, 6–10 June 2005, Barcelona, Spain, 2005, pp. 2587–2590.
[38] M. Calvo-Fernndez, J. E. Vega, M. A. Egido, E. Caamao-Martn, Spanish participation in the “solar decathlon2005”: design and simulation of the photovoltaic system (2005) 1958–1963.
[39] M. Castillo-Cagigal, E. Matallanas, A. Gutirrez, F. Monasterio-Huelin, E. Caama-Martn, D. Masa-Bote, J. Jimnez-Leube, Heterogeneous collaborative sensor network for electrical management of an automated house with PV energy, Sensors 11 (12) (2011) 11544–11559.
[40] K. Cho, B. van Merrienboer, Ç. Gülçehre, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder-decoder for statistical machine translation, CoRRabs/1406.1078(2014).
[41] I. Sutskever, O. Vinyals, Q.V. Le, Sequence to sequence learning with neural networks, CoRRabs/1409.3215(2014).
[42] K. Cho, B. van Merrienboer, D. Bahdanau, Y. Bengio, On the properties of neural machine translation: encoder-decoder approaches, arXiv:1409.1259 (2014).
[43] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, A. Lerer, Automatic differentiation in pytorch, NIPS-W, 2017.
[44] G. Hinton, Neural Networks for Machine Learning, Coursera, video lectures, 2012.